Cost-Sharing Methods for Scheduling Games under Uncertainty

GIORGOS CHRISTODOULOU, University of Liverpool
VASILIS GKATZELIS, Drexel University
ALKMINI SGOURITSA, University of Liverpool

We study the performance of cost-sharing protocols in a selfish scheduling setting with load-dependent cost functions. Previous work on selfish scheduling protocols has focused on two extreme models: omnipotent protocols that are aware of every machine and every job that is active at any given time, and oblivious protocols that are aware of nothing beyond the machine they control. The main focus of this paper is on a well-motivated middle-ground model of *resource-aware* protocols, which are aware of the set of machines that the system comprises, but unaware of what jobs are active at any given time. Apart from considering budget-balanced protocols, to which previous work was restricted, we augment the design space by also studying the extent to which *overcharging* can lead to improved performance.

We first show that, in the omnipotent model, overcharging enables us to enforce the optimal outcome as the unique equilibrium, which largely improves over the $\Theta(\log n)$-approximation of social welfare that can be obtained by budget-balanced protocols, even in their best equilibrium [32]. We then transition to the resource-aware model and provide price of anarchy (PoA) upper and lower bounds for different classes of cost functions. For concave cost functions, we provide a protocol with PoA of $1 + \epsilon$ for arbitrarily small $\epsilon > 0$. When the cost functions can be both convex and concave we construct an overcharging protocol that yields PoA≤ 2; a spectacular improvement over the bounds obtained for budget-balanced protocols, even in the omnipotent model. We complement our positive results with impossibility results for general increasing cost functions. We show that any resource-aware budget-balanced cost-sharing protocol has PoA of $\Theta(n)$ in this setting and, even if we use overcharging, no resource-aware protocol can achieve a PoA of $o(\sqrt{n})$.

CCS Concepts: • **Theory of computation → Quality of equilibria**;

Additional Key Words and Phrases: cost-sharing; price of anarchy; resource-aware protocols

1 INTRODUCTION

We consider a very basic resource allocation setting from a game theoretic point of view. There is a collection of m available machines and a set of n users, each of which needs access to one of the machines in order to process some task. For each user i there is a weight w_i associated with his task, that may represent its duration or its complexity; the higher the weight of a task, the higher the cost incurred by the machine that processes it. More precisely, each machine j is described by a cost function c_j that is weakly increasing in the total weight of the jobs assigned to it, and each machine's users need to cover the processing costs of their jobs.

The first author was supported by EPSRC EP/M008118/1 and Royal Society LT140046. The second author was supported by the National Science Foundation, under grants CCF-1216073, CCF-1161813, CCF-1408635. Part of this work took place while the authors were visiting the Simons Institute for the Theory of Computing.

XXXX-XXXX/2017/1-ART1 $15.00
https://doi.org/http://dx.doi.org/10.1145/3033274.3085151

From the perspective of a user in this setting, the goal is to minimize the cost that he needs to cover (be it monetary payment, delay, or some other type of inconvenience) for processing his job. Therefore, the decision of each user regarding which machine to utilize depends directly on how the induced cost of each machine is shared among its users, and different approaches regarding how the cost is to be shared may lead to vastly different outcomes. A *cost-sharing protocol* in this setting provides the rules regarding how the cost of each machine is to be shared among its users. Given such a protocol, the users strategically select which machine to use, taking both the protocol and the anticipated congestion into consideration. This forms a non-cooperative game whose equilibria can be quite inefficient, even for very natural cost-sharing protocols and extremely simple instances [2]. The focus of this paper is on the design of cost-sharing methods aiming to maximize the efficiency of the equilibria.

Motivated by the observation that such protocols can have a significant impact on the quality of the outcome, Chen et al. [8] were the first to address cost-sharing design questions. They focused on network cost-sharing, i.e., the case when the players can choose multiple machines but the cost functions are all constant. They gave a characterization of protocols that satisfy some natural axioms and they thoroughly studied their worst-case performance. Building on top of this work, von Falkenhausen and Harks [32] focused on the case where each player needs to choose a single machine or a matroid of machines, but allowed general cost functions as we also do in this work.

One essential way in which our work deviates from this previous work has to do with the informational constraints applied to the cost-sharing protocol. The information that the protocol is assumed to have access to plays a very important role in the extent to which efficient outcomes can be achieved. For instance, can the decisions of the protocol regarding how to distribute the cost of a machine depend on which other machines exist in the system? Can it depend on the set of users that are populating the system at each time? The more information the protocol possesses, the higher its power to reach high quality outcomes. Previous work focused on the following two informational assumptions.

- **Omnipotent protocols**[1]. The protocol of each machine has *full knowledge* of the instance. The cost-sharing decisions on that machine can depend arbitrarily on the state of the system: on the set of available machines and their cost functions, as well as on the set of active users and their weights.
- **Oblivious protocols**[2]. The protocol of each machine has *no knowledge* of the instance, except the set of users who chose to utilize that particular machine.

These assumptions lie at two extremes of the information spectrum. The former applies to a very static, or centralized, system where each machine always has access to up-to-date information regarding the users and the machines. The latter, on the other hand, is very pessimistic, assuming that the cost-sharing decisions of each machine need to be oblivious to the state of the system. Aiming for a middle-ground, we study a compromise between these extremes which is much more reasonable for many settings. Following the "adversarial" model of [12], we maintain the assumption that the cost-sharing protocol of a machine is aware of the other machines that are available in the system, but we restrict it to depend only on the particular machine's users.

- **Resource-aware protocols**. The protocol of each machine knows the set of available machines and their cost functions, but is aware only of the set of users who chose to utilize that particular machine.

[1]Previous work has also referred to these protocols as "non-uniform" protocols.
[2]Previous work has also referred to these protocols as "uniform" protocols.

This model is more realistic than the oblivious one when the machines of the system do not change often, which is the case in many scenarios. In that case, this information can be taken into consideration when designing or updating the cost-sharing protocol. For instance, if the machines correspond to computers in a static network, the designer could take the structure of the network into consideration when selecting which protocol to use. On the other hand, the set of active users in a system may change more often, as users arrive and depart. In fact, even if the machines had access to up-to-date information on the whole set of active users, protocols that depend on such dynamic information would complicate the user's decision, and could thus be less stable.

Hence, knowing the set of machines and their cost functions, the designer wishes to design a protocol that defines how the cost of each machine is to be shared among the users that utilize it, depending only on this particular set of users. In designing such protocols, the goal is to ensure that the worst-case equilibria of the induced game approximate the optimal social welfare within some small factor. We measure the performance of these protocols using the worst-case price of anarchy (PoA) measure, i.e., the ratio of the social welfare in the worst equilibrium over that in the optimal solution. Essentially, once the designer selects the protocol on each machine, then an adversary chooses the requested subset of players so that the PoA of the induced game is maximized.

Overcharging. A second important way in which our work deviates from previous work is the fact that we do consider protocols that use overcharging. Specifically, Chen et al. [8] and von Falkenhausen and Harks [32] restricted themselves to *budget-balanced* protocols, i.e., protocols such that the costs of the users using some machine add up to *exactly* the cost of the machine. Similarly to the framework of "coordination mechanisms" [9], we relax this assumption and seek to optimize the welfare via overcharging: although we maintain the restriction that the users need to pay at least as much as the cost that they create, in total, our protocols can also choose to charge the users additional costs in order to better align their incentives. It is important to note that, once we introduce increased costs in our protocols, we compare the performance of the equilibria in the induced game with the *increased* costs to the original optimal solution with the *initial* cost functions. Therefore, using this technique requires a careful trade-off between the benefit of improved incentives and the drawback of penalties suffered.

1.1 Our Results

We begin with a simple observation that demonstrates the power of overcharging for omnipotent protocols (Sec. 3). We show that, using overcharging, the designer can achieve the optimal outcome. In fact, no user actually suffers these extra charges in equilibrium, so this approach yields a PoA of 1 for arbitrary cost functions. This is in contrast to the $\Omega(\log n)$ bound obtained by budget-balanced omnipotent protocols, even for the best possible equilibria (price of stability) in [32].

We then move on to study resource-aware protocols, where our main results are the following.

- (Sec. 4) We provide a budget-balanced resource-aware cost-sharing protocol that achieves the optimal outcome, i.e., a PoA of 1, for functions with decreasing marginals (strictly concave functions). In contrast, we show that the PoA for oblivious protocols is lower bounded by $\Omega(n)$ for the same class of functions.
- (Sec. 5) We construct an overcharging resource-aware protocol that yields a PoA of at most 2 for the much more challenging case when both convex and concave cost functions appear in the system. This result holds only for unweighted players. For this class of functions, budget-balanced protocols, even omnipotent ones, are known to yield an $\Omega(\log n)$ approximation in their best equilibrium.
- (Sec. 6) We complement our positive results by proving lower bounds for resource-aware protocols. We show that, no policy can achieve a $o(\sqrt{n})$ PoA, even by the use of overcharging.

In the case of budget-balanced protocols we show that the respective bound is $\Theta(n)$. We stress that showing lower bounds for resource-aware mechanisms that allow overcharging is much more involved than the corresponding proofs for budget-balanced mechanisms, since the design space is considerably larger.

All our lower bounds hold for the unweighted case, so they immediately carry over to the weighted case. We should also point out that the lower bounds for resource-aware protocols hold also for oblivious protocols (but not vice versa). Finally, in Sec. 7 we provide results for the special case of two machines with general cost functions and unweighted players. We propose a protocol which does not fall into the class of the generalized weighted Shapley mechanisms [17], but still always possesses pure Nash equilibria. This protocol achieves a PoA of 2. We also show a PoA lower bound of 1.36 for all resource-aware protocols (with overcharging). Note that this lower bound applies also to the case of both convex and concave functions of Sec. 5, for which we achieved a PoA of 2.

1.2 Related Work

Cost-sharing protocols for resource selection games have been the focus of a lot of recent work [16, 17, 20, 24]. Harks and Miller [20] study the performance of several cost-sharing methods in a slightly modified setting, where each player declares a different demand for each resource. Marden and Wierman [24] studied various cost-sharing methods in a utility maximization model for the players, while Gopalakrishnan et al. [17] characterized the space of admissible cost-sharing methods as the set of generalized weighted Shapley values. Gkatzelis et al. [16] explored the space of cost-sharing methods and identified the Shapley value as the optimal one with respect to the price of anarchy for a class of general resource selection games.

Christodoulou et al. [10] studied network design games with constant cost functions under the Bayesian setting, where the position of the players' sources on the graph is drawn from a distribution over all vertices. They considered overcharging, where they could use any non-budget-balanced policy under the restriction of preserving budget-balance in all equilibria. In contrast to the scheduling games we study here, where the costs are functions of the machines' loads, in [10] the costs are functions of the set of players using a resource. Therefore, it turns out that their overcharging scheme is equivalent to the one that defines new cost (set) functions and then uses a budget-balanced protocol, similarly to our scheme.

An interesting family of resource selection games is the class of weighted congestion games. There has been a long line of work on these games focusing on the proportional sharing method, according to which the players share their joint cost in proportion to the size of their demands, [3, 6, 15, 18, 19, 25, 26, 31]. Proportional sharing does not always guarantee the existence of a pure Nash equilibrium, except for special cases, such as when the costs are quadratic or exponential functions of the load [14, 18]. Kollias and Roughgarden [23] were the first to propose the use of Shapley value based methods in congestion games in order to restore stability.

The impact of cost-sharing methods on the quality of equilibria has also been studied in other models: Moulin and Shenker [30] focused on participation games, while Moulin [29] and Mosk-Aoyama and Roughgarden [27] studied queueing games. Also, very closely related in spirit is previous work on coordination mechanisms, beginning with Christodoulou et al. [9] and subsequently in [1, 4, 5, 7, 11, 13, 21, 22]. Most work on coordination mechanisms concerns scheduling games and how the price of anarchy varies with the choice of local machine policies (i.e., the order in which to process jobs assigned to the same machine).

2 PRELIMINARIES

The games that we study comprise a set $N = \{1, \ldots, n\}$ of players, and a set $M = \{1, \ldots, m\}$ of available machines. Each player i needs to schedule a job to one of the machines and he can choose which machine to schedule it on; hence, the set of strategies corresponds to the set M. We consider "weighted" instances, where each agent may have a different weight $w_i \geq 1$, as well as "unweighted" instances, where $w_i = 1$ for every i. We denote the total weight of the players in the game by $W = \sum_{i \in N} w_i$. Each machine j is characterized by a cost function $c_j(\cdot)$ which is increasing in the total load on the machine and satisfies $c_j(0) = 0$.

Given a strategy profile (or schedule) $\mathbf{s} = (s_1, s_2, ..., s_n)$, let $S_j(\mathbf{s}) = \{i \in N : s_i = j\}$ be the set of players scheduling their jobs on machine j, and let $\ell_j(\mathbf{s}) = \sum_{i \in S_j(\mathbf{s})} w_i$ be the load on machine j. The cost of j in this schedule is $c_j(\ell_j(\mathbf{s}))$, and this cost needs to be covered by the set of players using this machine, $S_j(\mathbf{s})$. In this paper we design *cost-sharing methods*, i.e., protocols that decide how the cost of each machine will be distributed among its users. A cost-sharing protocol Ξ defines, at each schedule $\mathbf{s}$, a cost share $\xi_{ij}(\mathbf{s})$ for each $i \in N$ and $j \in M$. Since the machine that i uses, i.e., s_i, is implicitly part of $\mathbf{s}$, we may also denote this cost share as $\xi_i(\mathbf{s})$.

Our goal is to design policies that yield efficient outcomes in all games within a broader class. Formally, a class of scheduling games $\mathcal{G} = (\mathcal{N}, \mathcal{M}, \mathcal{C}, \Xi)$ comprises a universe of players $\mathcal{N}$, a universe of machines $\mathcal{M}$, whose cost functions are chosen from the set $\mathcal{C}$, and a cost sharing protocol Ξ. A game $G \in \mathcal{G}$ then consists of a set $M \subseteq \mathcal{M}$ of machines with cost functions from $\mathcal{C}$, a set of agents $N \subseteq \mathcal{N}$, and the cost sharing protocol Ξ.

Based on the information that the cost sharing protocol can depend on, previous work has focused on two extreme protocol design domains. On one extreme is the class of *oblivious protocols*, in which the cost shares $\xi_{ij}(\mathbf{s})$ for a machine j can depend only on the cost function $c_j(\cdot)$ of that machine and on the set $S_j(\mathbf{s})$ of players using j in $\mathbf{s}$. In other words, two machines with the same cost function are identical in terms of the way they share the cost, and their cost sharing is independent of what other machines or players are participating in the game at hand. On the other extreme, in an *omnipotent protocol* the cost shares $\xi_{ij}(\mathbf{s})$ for machine j can depend on *any* information regarding the game at hand. That is, apart from the set $S_j(\mathbf{s})$, the protocol is also aware of the set of participating players, N, the set of machines, M, and the cost functions of all these machines. In this paper we propose the class of *resource-aware protocols* that strike a balance between these two extremes. In particular, apart from the set of jobs $S_j(\mathbf{s})$, these protocols allow $\xi_{ij}(\mathbf{s})$ to depend on the set of machines M and their cost functions. They cannot, however, depend on the set of other jobs, $N \setminus S_j(\mathbf{s})$, that participate in the specific game.

A change of the cost-sharing protocol leads to a different class of games and, hence, to possibly very different outcomes. As a result, the efficiency of a game crucially depends on the choice of the protocol. In evaluating the performance of a cost-sharing protocol, we measure the quality of the pure Nash equilibria that arise in the games that it induces. A strategy vector $\mathbf{s}$ is a *pure Nash equilibrium* (PNE) of a game G if for every player $i \in S_j(\mathbf{s})$, and every other strategy $s_i' \in M$

$$\xi_i(\mathbf{s}) \;=\; \xi_i(s_i, \mathbf{s}_{-i}) \;\leq\; \xi_i(s_i', \mathbf{s}_{-i}).$$

As a measure of efficiency of a schedule $\mathbf{s}$, we use the total cost $C(\mathbf{s}) = \sum_{j \in M} c_j(\ell_j(\mathbf{s}))$, and we quantify the performance of the cost-sharing protocol using the price of anarchy metric. Given a cost sharing protocol Ξ, the *price of anarchy* (PoA) of the induced class of games $\mathcal{G} = (\mathcal{N}, \mathcal{M}, \mathcal{C}, \Xi)$ is the worst-case ratio of equilibrium cost to optimal cost over all games in $\mathcal{G}$. That is, if $E(G)$ is the set of pure Nash equilibria of the game G, and $F(G)$ the set of schedules of G, then

$$\mathrm{PoA}(\mathcal{G}) = \sup_{G \in \mathcal{G}} \frac{\max_{\mathbf{s} \in \mathrm{E}(G)} C(\mathbf{s})}{\min_{\mathbf{s}^* \in F(G)} C(\mathbf{s}^*)}.$$

We say that a cost sharing protocol is *budget-balanced* if the cost that it distributes to the users of each machine adds up to exactly the cost of the machine, i.e., $\forall j \in M$, $c_j(\ell_j(\mathbf{s})) = \sum_{i \in S_j(\mathbf{s})} \xi_{ij}(\mathbf{s})$. A protocol is *stable* within a class of games $\mathcal{G}$ that it induces, if every $G \in \mathcal{G}$ always possesses at least one PNE. In this paper, we restrict our attention to protocols that are stable.

If the mechanism is not budget-balanced, it may define cost functions such that $\hat{c}_j(\ell) > c_j(\ell)$ for some loads ℓ. As a result, the social cost of a given schedule $\mathbf{s}$ may be increased from $C(\mathbf{s})$ to $\hat{C}(\mathbf{s})$. In these mechanisms, we measure the quality of the equilibria using the new costs, but we compare their performance to the optimal solutions based on the original cost functions. If $\hat{G}$ is the game induced by the new cost functions $\hat{c}_j(\ell)$, and $E(\hat{G})$ is the set of pure Nash equilibria of $\hat{G}$,

$$\text{PoA}(\mathcal{G}) = \sup_{G \in \mathcal{G}} \frac{\max_{\mathbf{s} \in \text{E}(\hat{G})} \hat{C}(\mathbf{s})}{\min_{\mathbf{s}^* \in F(G)} C(\mathbf{s}^*)}.$$

Some of our results focus on the games induced by specific classes of functions C. In particular, a large part of this paper focuses on convex functions (exhibiting non-decreasing marginal costs), concave cost functions (exhibiting non-increasing marginal costs), or combinations of the two. Apart from these functions, a class of functions that plays an important role is that of *capacitated constant* cost functions. That is, functions such that $c(\ell) = c_1$ when $\ell \leq c_2$ and $c(\ell) = \infty$ when $\ell > c_2$, for positive constants c_1 and c_2. These functions correspond to a machine whose cost is constant as long as the load does not exceed its capacity. It is important to point out that these functions are not concave, since our cost functions satisfy $c_j(0) = 0$. In fact, as we show, this class of functions poses significant obstacles to cost-sharing protocols.

2.1 Related Results for Machines with Convex Functions

We now briefly consider the class of games where all the cost functions are convex, and we present a simple oblivious protocol proposed by Moulin [28]. In Section 5 we use this protocol as a component of our own protocol for the larger class of instances that include both convex and concave functions. Moulin's *incremental cost-sharing protocol*, like many of the known cost-sharing protocols in the related work, uses a global ordering over the universe of players in deciding how to distribute the cost. Some of our mechanisms use this ordering as well, so we provide a definition below.

Definition 2.1. For a class of games $\mathcal{G} = (\mathcal{N}, \mathcal{M}, C, \Xi)$, the *global ordering* π is an ordering of all the players in $\mathcal{N}$ in a non-increasing order with respect to their weight.

Definition 2.2. Given a schedule $\mathbf{s}$ and a machine j, the *prior load*, $\ell_j^{<i}(\mathbf{s})$, for a player $i \in S_j(\mathbf{s})$ is the load on j in $\mathbf{s}$ due to jobs that precede i in π, i.e.,

$$\ell_j^{<i}(\mathbf{s}) = \sum_{k \in S_j(\mathbf{s}): \pi(k) < \pi(i)} w_k.$$

In games with convex cost functions and unweighted jobs the optimal assignment can be computed via a greedy algorithm that assigns each job to the machine where it causes the smallest marginal contribution to the cost function. The incremental cost-sharing protocol charges each user its marginal contribution with respect to the prior load.

Definition 2.3. The *incremental cost-sharing protocol* defines the cost-share of each $i \in S_j(\mathbf{s})$ to be its marginal contribution if only players preceding it in π were using machine j.

$$\xi_{ij}(\mathbf{s}) = c_j(\ell_j^{<i}(\mathbf{s}) + w_i) - c_j(\ell_j^{<i}(\mathbf{s}))$$

The incremental cost-sharing protocol is oblivious, budget-balanced, and stable [28]. Furthermore, when the players are unweighted, it achieves a PoA of 1.

Theorem 2.4. *[28] For convex cost functions and unweighted jobs, the incremental cost-sharing protocol induces games with PoA = 1.*

Although the equilibria of this protocol are optimal when the players are unweighted, for weighted jobs, the incremental cost-sharing protocol induces games with unbounded PoA [32].

3 OMNIPOTENT PROTOCOLS

In this section, we demonstrate the power of overcharging for omnipotent protocols. We show that overcharging can dramatically decrease the PoA of the game. In particular, even for unweighted instances, no budget-balanced omnipotent mechanism can achieve a PoA better than the harmonic number $\mathcal{H}_n = \Omega(\log n)$. In fact, this bound applies even to the best equilibria of the induced games (the price of stability). However, a simple overcharging scheme enables us to reach a PoA of 1 for weighted instances.

Given any problem instance, using an optimal schedule $\mathbf{s}^*$, following the approach of [32], we define the set of "foreign" players of machine j in an outcome $\mathbf{s}$ as the subset of players using j in $\mathbf{s}$ but not in $\mathbf{s}^*$. We then define an efficient omnipotent protocol that overcharges foreign players.

Definition 3.1. The set of *foreign players* of j in $\mathbf{s}$ is $F_j(\mathbf{s}) := \{i \in S_j(\mathbf{s}) \setminus S_j(\mathbf{s}^*)\}$.

Definition 3.2. Given some large constant $D > \sum_{j\in M} c_j(W)$ and the players' ordering π, the *omnipotent overcharging protocol* defines the *modified* costs $\hat{c}_j(\cdot)$ of the machines and the cost shares $\xi_{ij}(\cdot)$ as follows:

$$\hat{c}_j(\ell) = \begin{cases} c_j(\ell), & \text{if } \ell \leq \sum_{i\in S_j(\mathbf{s}^*)} w_i; \\ D, & \text{otherwise,} \end{cases}$$

and

$$\xi_{ij}(\mathbf{s}) = \begin{cases} \frac{w_i}{\sum_{k\in S_j(\mathbf{s})} w_k} \hat{c}_j(\ell_j(\mathbf{s})), & \text{if } F_j(\mathbf{s}) = \emptyset; \\ \hat{c}_j(\ell_j(\mathbf{s})), & \text{if } F_j(\mathbf{s}) \neq \emptyset \text{ and } i = \arg\min_k \{\pi(k) | k \in F_j(\mathbf{s})\}; \\ 0, & \text{otherwise.} \end{cases}$$

This protocol keeps the costs of each machine unaffected, unless the total load on this machine exceeds its total load in $\mathbf{s}^*$, in which case the cost jumps up to some large constant D. If there are no foreign players on a machine j, its cost is divided among its users in a proportional manner. Otherwise, all of its cost, which need not be D, is charged to a single foreign player (the one that appears earliest in π) and every other user suffers no cost.

The next lemma shows that this protocol is stable and that every equilibrium corresponds to an optimal schedule. Furthermore, we show that, despite the extra charges this protocol introduces, no charges are used in any equilibrium, leading to a PoA of 1.

Lemma 3.3. *The omnipotent overcharging protocol is stable and it has PoA* 1.

Proof. We first show that (i) $\mathbf{s}^*$ is a Nash equilibrium of the game induced by the omnipotent overcharging protocol, and (ii) for any other Nash Equilibrium $\tilde{\mathbf{s}}$, it is $\hat{C}(\tilde{\mathbf{s}}) \leq C(\mathbf{s}^*)$.

To show (i) note that, since there are no foreign players on any machine j at $\mathbf{s}^*$, then $\hat{c}_j(\ell_j(\mathbf{s}^*)) = c_j(\ell_j(\mathbf{s}^*))$. Hence, in $\mathbf{s}^*$, the cost of every player is $\xi_i(\mathbf{s}^*) \leq c_j(\ell_j(\mathbf{s}^*)) \leq D$. If some player i can improve his cost by unilaterally deviating from s_i^* to some other machine j, his new cost should be less than D. But, this deviation makes i the *only foreign player* on that machine, since every other player i' is still using $s_{i'}^*$. The cost for this machine after this deviation is $\hat{c}_j(\ell_j(\mathbf{s}^*) + w_i) = D$, and all of this cost is charged to i, i.e., $\xi_i(j, \mathbf{s}^*_{-i}) = D$, proving that $\mathbf{s}^*$ is a Nash equilibrium.

To see (ii), let $\tilde{\mathbf{s}}$ be a Nash equilibrium of the induced game. If there is a machine with $\ell_j(\tilde{\mathbf{s}}) > \ell_j(\mathbf{s}^*)$, then clearly there is a "foreign" player i using this machine that pays a cost of D. But, if i were to

deviate to machine $j' = s_i^*$, his cost would be less than D, contradicting that $\tilde{\mathbf{s}}$ is an equilibrium. To verify this fact note that if $F_{j'}(\tilde{\mathbf{s}}) \neq \emptyset$, then i would suffer a cost of 0 and if $F_{j'}(\tilde{\mathbf{s}}) = \emptyset$, then i would suffer a cost of $\xi_i(j', \tilde{\mathbf{s}}_{-i}) \leq c_{j'}(\mathbf{s}^*) < D$. Therefore, for any equilibrium $\tilde{\mathbf{s}}$ and every machine j, the loads are $\ell_j(\tilde{\mathbf{s}}) \leq \ell_j(\mathbf{s}^*)$, which implies $\hat{C}(\tilde{\mathbf{s}}) \leq \hat{C}(\mathbf{s}^*) = C(\mathbf{s}^*)$. □

4 MACHINES WITH CONCAVE COST FUNCTIONS

In Section 2 we showed that, for the class of convex cost functions, there exists a simple oblivious, stable, and budget-balanced mechanism with PoA of 1. We begin this section by showing that this is far from true when it comes to concave functions. In particular, Theorem 4.1 shows that the PoA of any oblivious, stable, and budget-balanced mechanism grows linearly with n, even for strictly concave functions. We then complement this result with a resource-aware, stable, and budget-balanced mechanism that achieves a PoA of 1 for strictly concave cost functions. We conclude this section by showing how we can apply a negligible amount of overcharging in order to extend our positive result to general (non-strictly) concave cost functions. In particular, we use overcharging to transform the function into a strictly concave one and then apply our resource-aware protocol over the new cost functions, leading to a PoA of $1 + \varepsilon$ for an arbitrarily small constant $\varepsilon > 0$.

4.1 Oblivious Budget-Balanced Protocols

Theorem 4.1. *Any oblivious, stable, and budget-balanced cost-sharing policy has PoA of at least n, even for the class of strictly concave cost functions and in the unweighted setting.*

Proof. Aiming for a contradiction, assume that there exists some oblivious, stable, and budget-balanced policy with PoA of at most $n - \delta$ for some constant $\delta > 0$. Consider the outcomes of this policy for an instance comprising n unweighted agents and n machines with cost functions $c_j(\ell) = \ell + \varepsilon - \varepsilon/\ell$, for some arbitrarily small constant $\varepsilon > 0$.

Since this policy is stable, it is guaranteed to possess an equilibrium for this instance. We show that in any such equilibrium $\mathbf{s}$, every player needs to pay a cost of at most 1. If, in $\mathbf{s}$, each agent is matched to a different machine, this is obvious, since the policy is budget-balanced, and $c_j(1) = 1$. Also, if more than one agent use some machine j in $\mathbf{s}$, the cost share of each one of these agents should be at most 1, otherwise, any agent with a cost share greater than 1 would benefit from a unilateral deviation to one of the "empty" machines (there is always one, since the number of machines and agents is the same).

Now, consider the outcomes of this same policy in an instance where, apart from the same set of jobs and same set of machines, we introduce an extra machine $n + 1$ with cost $c_{n+1}(\ell) = 1 + \varepsilon - \varepsilon/\ell$. Since the protocol is oblivious, the cost functions of the old machines are not affected by the existence of this new machine, and any equilibrium $\mathbf{s}$ of the previous instance, remains an equilibrium in the new instance. To verify this fact, note that the cost of the new machine is at least 1 and, as we showed above, the cost of every player in $\mathbf{s}$ is at most 1. But, any such equilibrium has social cost at least $n - \varepsilon$, when the optimal solution would be for all agents to use the last machine with social cost of at most $1 + \varepsilon$. Hence, for small enough values of ε, this protocol's PoA is greater than $n - \delta$, leading to a contradiction. □

4.2 Resource-Aware Budget-Balanced Protocol

We now present the main result of this section, which is an efficient resource-aware protocol that is both stable and budget-balanced. We begin by assuming that the cost functions are *strictly* concave and then also address the general concave functions case.

The result of Theorem 4.1 implies that, in order to achieve a good PoA, a resource-aware protocol would need to take advantage of the additional information that it has access to, compared to

an oblivious protocol. In particular, this protocol will need to use the information regarding the available machines and their cost functions in such a way that the agents are incentivized to avoid inefficient outcomes. But, what information regarding the machines of the system should such a protocol use, and how can we describe such a protocol that works for *all* possible instances, in a concise way? Also, note that the protocol is not going to be aware of what subset of agents is active, so it needs to work for *any* such subset. Before presenting our protocol, we begin with a few definitions and observations.

Definition 4.2. Given a schedule, $\mathbf{s}$, and a global ordering of the players, π, the *highest priority player* of some non-empty machine j in $\mathbf{s}$, denoted by $h_j(\mathbf{s})$, or simply h_j, is the player $i \in S_j(\mathbf{s})$ with the highest priority in π, i.e.,

$$h_j(\mathbf{s}) = \arg \min_{i' \in S_j(\mathbf{s})} \{\pi(i')\}.$$

The following definition provides the information regarding the set of machines in the system that our resource-aware protocol uses in order to outperform any oblivious protocol.

Definition 4.3. Given a set of machines and some load ℓ, let $\phi(\ell) = \min_{j \in M} c_j(\ell)$ be the smallest cost over all the machines for serving this load, and note that this function $\phi : \mathbb{R}_+ \to \mathbb{R}_+$ is concave as minimum of concave functions. Also, let $X_{\min}(\ell)$ be the set of machines with the smallest cost when their load is ℓ, i.e., $X_{\min}(\ell) = \arg\min_{j \in M}\{c_j(\ell)\}$.

Remark 4.4. *If the total load in the game is W, then an optimal allocation assigns all the load to a machine $j^* \in X_{\min}(W)$. To verify this fact note that, due to the concavity of $\phi(\cdot)$, for any profile $\mathbf{s}$,*

$$\sum_{j \in M} c_j(\ell_j(\mathbf{s})) \;\geq\; \sum_{j \in M} \phi(\ell_j(\mathbf{s})) \;\geq\; \phi\left(\sum_{j \in M} \ell_j(\mathbf{s})\right) \;=\; \phi(W) \;=\; c_{j^*}(W).$$

We now define the concave cost-sharing protocol. In its description, for notational simplicity we have dropped the dependence of $h_j(\mathbf{s})$ on $\mathbf{s}$, thus replacing it with h_j.

Definition 4.5. According to the *concave cost-sharing protocol*, given a machine j and a schedule $\mathbf{s}$, the cost share of player $i \in S_j(\mathbf{s})$ is

$$\xi_{ij}(\mathbf{s}) = \begin{cases} c_j(\ell_j(\mathbf{s})) & \text{if } j \notin X_{\min}(\ell_j(\mathbf{s})) \text{ and } i = h_j \\ 0 & \text{if } j \notin X_{\min}(\ell_j(\mathbf{s})) \text{ and } i \neq h_j \\ \phi(w_i) & \text{if } j \in X_{\min}(\ell_j(\mathbf{s})) \text{ and } i = h_j \\ w_i \frac{c_j(\ell_j(\mathbf{s})) - \phi(w_{h_j})}{\ell_j(\mathbf{s}) - w_{h_j}} & \text{if } j \in X_{\min}(\ell_j(\mathbf{s})) \text{ and } i \neq h_j \end{cases}$$

Note that, according to Remark 4.4, the optimal solution can change radically, depending on the set of jobs that are active in the system. Being unaware of which jobs are active, a resource-aware cost-sharing mechanism is therefore unable to guess which machine it should be incentivizing the players to use. In the absence of this information, our cost-sharing policy essentially treats the players that use a machine j as if they are the only active ones in the system. Hence, if the load of this machine is $\ell_j(\mathbf{s})$, but $j \notin X_{\min}(\ell_j(\mathbf{s}))$, then $\mathbf{s}$ should not be permitted as an equilibrium, and the protocol incentivizes the $h_j(\mathbf{s})$ to leave the machine by charging the whole cost to him.

Of course, this high level intuition does not go far enough in explaining why the protocol actually works. For instance, one could come up with schedules $\mathbf{s}$ such that two (or more) different machines $j1$ and $j2$ have positive loads that satisfy $j1 \in X_{\min}(\ell_{j1}(\mathbf{s}))$ and $j2 \in X_{\min}(\ell_{j2}(\mathbf{s}))$. In such schedules, the myopic view of the protocol for each one of these machines is that $\mathbf{s}$ may as well be optimal. Therefore, the most interesting part of this protocol is the way in which it distributes the costs

in these circumstances, using the $\phi(w_i)$ function, which ensures that such "locally optimal" but globally inefficient schedules will not be equilibrium outcomes.

Since the protocol depends only on the cost functions of the machines, but not on the total number of agents and their weights, it is a valid resource-aware protocol. As we show below, it is also budget-balanced, stable, and its PoA is 1.

LEMMA 4.6. *The concave cost-sharing protocol is budget-balanced.*

PROOF. Given some problem instance, we show that for any schedule $\mathbf{s}$ and any machine j, the sum of the cost shares of the players in $S_j(\mathbf{s})$ adds up to exactly $c_j(\ell_j(\mathbf{s}))$. If $j \notin X_{\min}(\ell_j(\mathbf{s}))$, then every player $i \neq h_j(\mathbf{s})$ has a cost of 0, and player $h_j(\mathbf{s})$ suffers the whole cost $c_j(\ell_j(\mathbf{s}))$. On the other hand, if $j \in X_{\min}(\ell_j(\mathbf{s}))$, the cost of $h_j(\mathbf{s})$ is $\phi(w_{h_j})$, while the total cost of all the other players is

$$\sum_{i \in S_j(\mathbf{s}) \setminus \{h_j\}} w_i \frac{c_j(\ell_j(\mathbf{s})) - \phi(w_{h_j})}{\ell_j(\mathbf{s}) - w_{h_j}} = (\ell_j(\mathbf{s}) - w_{h_j}) \frac{c_j(\ell_j(\mathbf{s})) - \phi(w_{h_j})}{\ell_j(\mathbf{s}) - w_{h_j}} = c_j(\ell_j(\mathbf{s})) - \phi(w_{h_j}).$$

□

LEMMA 4.7. *The concave cost-sharing protocol is stable.*

PROOF. To prove the stability of this protocol, we show that for any problem instance, the schedule $\mathbf{s}$ in which every player uses the same machine $j \in X_{\min}(W)$ is a Nash equilibrium.

In this schedule, the cost share of the highest priority player $i = h_j(\mathbf{s})$ is $\phi(w_i)$. Also, by Lemma 4.8, the cost share of every other player i is strictly less than $\phi(w_i)$. Therefore, every player i's cost in $\mathbf{s}$ is at most $\phi(w_i)$. But, if some player i were to deviate to another machine j', his new cost would be $c_{j'}(w_i) \geq \phi(w_i)$, so nobody has an incentive to deviate. □

The following lemma will help to prove the PoA bound of Theorem 4.9. It argues that for any machine j with load ℓ and $j \in X_{\min}(\ell)$, the cost-share of any agent $i \neq h_j(\mathbf{s})$ using j is strictly less than $\phi(w_i)$.

LEMMA 4.8. *At any profile $\mathbf{s}$ and every machine $j \in X_{\min}(\ell_j(\mathbf{s}))$, the concave cost-sharing protocol assigns to every $i \in S_j(\mathbf{s}) \setminus \{h_j(\mathbf{s})\}$ a cost share $\xi_{ij}(\mathbf{s}) < \phi(w_i)$.*

PROOF. Since $j \in X_{\min}(\ell_j(\mathbf{s}))$, the cost on j satisfies $c_j(\ell_j(\mathbf{s})) = \phi(\ell_j(\mathbf{s}))$. Therefore, for every player i who is not highest priority on j, i.e., $i \neq h_j$,

$$\xi_{ij}(\mathbf{s}) = w_i \frac{c_j(\ell_j(\mathbf{s})) - \phi(w_{h_j})}{\ell_j(\mathbf{s}) - w_{h_j}} = w_i \frac{\phi(\ell_j(\mathbf{s})) - \phi(w_{h_j})}{\ell_j(\mathbf{s}) - w_{h_j}} \leq w_i \frac{\phi(\ell_j(\mathbf{s}) - w_{h_j})}{\ell_j(\mathbf{s}) - w_{h_j}}.$$

Note that the denominator $\ell_j(\mathbf{s}) - w_{h_j}$ is always positive, since there is at least one more job $i \neq h_j$ using machine j in $\mathbf{s}$. In fact, $\ell_j(\mathbf{s}) - w_{h_j} \geq w_i$ so, using the fact that the cost functions are strictly concave, the previous inequality implies

$$\xi_{ij}(\mathbf{s}) \leq w_i \frac{\phi(\ell_j(\mathbf{s}) - w_{h_j})}{\ell_j(\mathbf{s}) - w_{h_j}} < \phi(w_i).$$

□

THEOREM 4.9. *The concave cost-sharing protocol has PoA = 1.*

PROOF. According to Remark 4.4, the schedule $\mathbf{s}$ in which all the players use a machine $j \in X_{\min}(W)$ is optimal. We now show that the social cost of any Nash equilibrium also equals $c_j(W)$. Aiming for a contradiction, assume that there exists some Nash equilibrium $\mathbf{s}'$ with cost greater than $c_j(W)$.

If, in $\mathbf{s}'$, every player uses the same machine $j' \notin X_{\min}(W)$, then the total cost is $c_{j'}(W) > c_j(W)$. If i is the highest priority player among them, then he pays a share equal to $c_{j'}(W)$. But, if this player deviated to machine j, his cost would become $c_j(w_i) \leq c_j(W) < c_{j'}(W)$, so this cannot be an equilibrium.

If, on the other hand, two or more machines are being used in $\mathbf{s}'$, let a and b be two of these machines, and let α and β be the corresponding highest priority players in $\mathbf{s}'$. Without loss of generality, assume that α has higher global priority than β, i.e., $\pi(\alpha) < \pi(\beta)$. According to the definition of the protocol, if $b \in X_{\min}(\ell_b(\mathbf{s}'))$, the cost share of β in $\mathbf{s}'$ is $\xi_\beta(\mathbf{s}') = \phi(w_\beta)$ and, if $b \notin X_{\min}(\ell_b(\mathbf{s}'))$ it is

$$\xi_\beta(\mathbf{s}') = c_b(\ell_b(\mathbf{s}')) \geq c_b(w_\beta) \geq \phi(w_\beta).$$

Therefore, in both cases, $\xi_\beta(\mathbf{s}') \geq \phi(w_\beta)$. But, if player β deviates to machine a, he is not the highest priority player on that machine (player α is), so his cost is less than $\phi(w_\beta)$. To verify this fact, we consider two different scenarios depending on whether, after this deviation, $a \in X_{\min}(\ell_a(\mathbf{s}') + w_\beta)$ or $a \notin X_{\min}(\ell_a(\mathbf{s}') + w_\beta)$. In the former case, Lemma 4.8 implies that the cost of β after the deviation is less than $\phi(w_\beta)$. In the latter, according to the definition of the protocol, the cost of β is 0. Therefore this cannot be an equilibrium either, which conclude the proof. □

4.3 General Concave Cost Functions

Since the positive result of the previous section focused on strictly concave functions, we now show how we can leverage overcharging in order to transform any weakly concave cost function into a strictly concave one. This way, we reduce this problem to the one we solved above and, as we show, the loss in PoA can be arbitrarily small. In particular, by overcharging, we define a cost-sharing protocol with PoA = $1 + \varepsilon$, where $\varepsilon > 0$ is an arbitrarily small constant.

The main observation is that we can implement this transformation by updating each $c_j(\ell)$ to $\hat{c}_j(\ell) = c_j(\ell) + \varepsilon/\ell$. Once we have done this, the $\hat{c}_j(\ell)$ is strictly concave, and the cost difference between $\hat{c}_j(\ell)$ and $c_j(\ell)$ is no more than ε. Although this overcharging may affect the set of equilibrium outcomes, the social cost of the optimal schedule will not be affected by more than some insignificant constant. Therefore, the PoA of this game will be arbitrarily close to 1, as ϵ goes to zero.

5 MACHINES WITH A MIX OF CONCAVE AND CONVEX COST FUNCTIONS

Unlike the previous two sections, the rest of the paper focuses exclusively on instances with unweighted jobs. We now consider the case where the cost function of each machine is either a concave or a convex function. If we focus on budget-balanced cost-sharing policies, the price of stability is known to be $\Omega(\log n)$ *even for omnipotent protocols* [32]. In stark contrast to this fact, we propose a non-budget-balanced resource-aware policy whose PoA is at most 2. Once again, this demonstrates the power of overcharging. In Section 7.2 we complement this positive result by proving that no resource-aware protocol can achieve a PoA better than 1.36 even for instances with just two machines with convex and concave cost functions, and even if overcharging is used.

5.1 Budget-Balanced Cost-Sharing

As shown in [32], even in the omnipotent case, where the designer has full knowledge of the instance, and even the *best* Nash equilibrium can be a $\Omega(\log n)$ factor away of the optimal cost. We describe here the lower bound for completeness.

Lower bound for budget balanced protocols. Consider the load balancing game of $m + 1$ machines, such that each machine j of the first m has a cost function $c_j(\ell)$ with $c_j(1) = 1/j$ and $c_j(\ell) = \infty$ for $\ell > 1$. The last machine has a cost of $c_{m+1}(\ell) = 1 + \varepsilon$ for all ℓ. Let m be the number of agents that

require to be served (we can even assume that this is known to the designer). Suppose now that $m' \geq 1$ of agents are served by the last machine. By budget-balance, there is at least one with share more that $1/m'$, whereas from the first m machines there should be at least one empty with cost at most $1/m'$ (there are $m - m' + 1$ machines with cost at most $1/m'$). So, such an outcome cannot be stable. Therefore, the only outcome that may be stable with bounded cost is for every agent to use a distinct machine from the m first. This results to a PoA and PoS of $\Omega(\log m)$.

5.2 Cost-Sharing with Overcharging

Let M_v, M_c be the subset of machines with convex and concave cost functions, respectively. Also, let ξ^{v} denote the incremental cost-sharing protocol defined in Sec. 2.1 for convex functions and ξ^{c} denote the protocol of Sec. 4 for concave functions. We now define a protocol that combines them.

Definition 5.1 (Convex-Concave (VC) cost-sharing protocol). Using the global order π of the players, we define the Ξ^{vc} as follows:

$$\xi_{ij}^{\text{vc}}(\mathbf{s}) = \begin{cases} \xi_{ij}^{\text{v}}(\mathbf{s}) & \text{if } j \in M_v \\ \xi_{ij}^{\text{c}}(\mathbf{s}) & \text{if } j \in M_c \end{cases}$$

where for ϕ and $X_{\min}$ (used to define ξ^{c}) are defined only over the set of concave functions.

Lemma 5.2. *The Convex-Concave cost-sharing protocol is stable.*

Proof. Let $\phi(\ell) = \min_{j \in M_c} c_j(\ell)$ be the minimum cost incurred by load ℓ, over all concave machines. Without loss of generality we assume that the identity of each player matches her position according to π, i.e. $\pi(i) = i$.

We define a profile $\mathbf{s}$ which is a pure Nash equilibrium. The first k (possibly zero) players with respect to the global ordering π, use only convex functions from M_v. The condition that needs to be satisfied is that k is the maximum integer such that $\xi_k^v(\mathbf{s}) \leq \phi(1)$. The rest of the players use the concave machine with the minimum cost for $n - k$ load, i.e. machine $X_{\min}(n - k)$.

If $k = 0$ (all players use concave machines), then as shown in Section 4, each player pays a share of at most $\phi(1)$, while for any $j \in M_v$ $c_j(1) > \phi(1)$. If $k = n$, (all players use convex machines), $\mathbf{s}$ is an equilibrium by the analysis of Section 2.1. Notice that each player pays share of at most $\phi(1)$ (by definition of k), so no player will deviate to a concave machine.

It is clear why a player will not switch from a convex to a convex and from a concave to a concave machine, by analysis similar to Sections 2.1 and 4. Now, observe that a player that uses a convex machine precedes, according to π, any player using a concave machine. Therefore, if she switches to any concave machine, she will pay at least $\phi(1)$. Moreover, recall that each player that uses the concave machine $X_{\min}(n - k)$, pays a share of at most $\phi(1)$, while if she deviates to a convex machine, she will be the last according to π, and will pay the next marginal cost, which exceeds $\phi(1)$, by definition of k. □

The following Lemma reveals a useful structural property of optimal schedules when the cost functions are either convex or concave. Due to space constraints, the proof of this Lemma is deferred to the full version of the paper.

Lemma 5.3. *If for n players some optimal schedule assigns $n_c > 0$ agents to concave machines and n_v agents to convex machines, then for $n' > n$ players there is an optimal schedule such that the number of players assigned to convex machines is at most n_v.*

Using the VC cost-sharing protocol, which is budget-balanced, we now define the VC-Overcharing protocol, which combines the VC protocol with appropriate overcharging to circumvent the limitations of budget-balanced protocols shown above. Given a set of machines, let $n^{\max}$ be the maximum

number of players such that, when $n \leq n^{\max}$, there is no optimal schedule $OPT(n)$ of n unweighted jobs to these machines that uses a concave machine. Also, let $n_j^{\max}$ be the number of players assigned to machine $j \in M_v$ in the $OPT(n^{\max})$ allocation.

Definition 5.4 (VC-Overcharging cost-sharing protocol). Apply the VC cost-sharing protocol to cost functions $\hat{c}_j(\ell)$ such that $\hat{c}_j(\ell) = c_j(\ell)$, unless $j \in M_v$ and $\ell \geq n_j^{max}$, in which case

$$\hat{c}_j(\ell) = \max\left\{\max_{j' \in M_c} c_{j'}(1)\ ,\ c_j(\ell)\right\}.$$

Remark 5.5. *The minimum number of players for which at least one concave machine is used by the optimum allocation is $n^{max}+1$. Moreover, by Lemma 5.3 for any $n > n^{max}$ there is an optimal schedule that assigns at most n^{max} players in total to convex machines. Therefore the optimal cost is not affected by the overcharging of the VC-Overcharging protocol. Furthermore, the arguments of Lemma 5.2 can be applied to show that the VC-Overcharging protocol is stable as well.*

Theorem 5.6. *For instances with concave and convex cost functions the VC-Overcharging protocol yields PoA at most* 2.

Proof. Recall that M_c, M_v are the sets of concave and convex machines respectively. Further, let $\phi_c(\ell)$ and $\phi_v(\ell)$ be the cost of optimally assigning ℓ players to only concave or only convex machines, respectively. Observe that ϕ_c, and ϕ_v are concave and convex respectively.

Let $\mathbf{s}$ be any Nash equilibrium, and $\mathbf{s}^*$ be the optimal allocation with at most n^{max} players in convex machines; the existence of such an optimal allocation is guaranteed by Remark 5.5. Let $n_c = \sum_{j \in M_c} \ell_j(\mathbf{s})$, and $n_v = \sum_{j \in M_v} \ell_j(\mathbf{s})$ be the total number of players using concave and convex machines, respectively, in the Nash equilibrium, and n_c^*, n_v^* the same numbers w.r.t $\mathbf{s}^*$.

Claim 5.7. *The total number of players using concave machines in $\mathbf{s}^*$ is at least as high as in $\mathbf{s}$, i.e., $n_c^* \geq n_c$, and hence $n_v^* \leq n_v$.*

Proof. Aiming for a contradiction, assume that $n_c^* < n_c$ and $n_v^* > n_v$. By optimality of n_c^*, n_v^*,

$$\phi_v(n_v^*) - \phi_v(n_v) < \phi_c(n_c) - \phi_c(n_c^*). \tag{1}$$

Let i^* be the smallest player w.r.t π that uses a concave machine in $\mathbf{s}$, and let j^* be the index of that machine (recall that all players that use concave machines, are assigned to j^*). Notice that there is such a player since by assumption $n_c > n_c^*$.

The share of i^* is $\xi_{i^*}(\mathbf{s}) = \xi_{i^*j^*}(\mathbf{s}) = \phi_c(1)$. Using Inequality (1), the fact that $n_v < n_v^*$, as well as convexity and concavity of ϕ_v and ϕ_c, respectively, we get

$$\phi_v(n_v+1) - \phi_v(n_v) \ \leq\ \frac{\phi_v(n_v^*) - \phi_v(n_v)}{n_v^* - n_v} \ <\ \frac{\phi_c(n_c) - \phi_c(n_c^*)}{n_c - n_c^*} \ \leq\ \phi_c(1) = \xi_{i^*}(\mathbf{s}).$$

So, player i^* has a reason to deviate to a convex machine, which contradicts the assumption that $\mathbf{s}$ is a Nash equilibrium. □

If $n \leq n^{max}$, then the the optimal cost is $\phi_v(n)$, i.e. $n_v^* = n$, based on the definition of n^{max}. By Claim 5.7 it is $n = n_v^* \leq n_v$, so trivially, $n = n_v$, and the PoA is 1.

Otherwise, based on the definition of n^{max} and on the VC-Overcharging cost-sharing protocol, both the optimal allocation and the Nash equilibrium assign at most n^{max} players to the convex machines. Therefore,

$$\begin{aligned} C(\mathbf{s}) \ &= \ \phi_c(n_c) + \phi_v(n_v) \ \leq\ \phi_c(n_c) + \phi_v(n_v^*) + \phi_c(n_v - n_v^*) \\ &= \ \phi_c(n_c) + \phi_v(n_v^*) + \phi_c(n_c^* - n_c) \ \leq\ \phi_c(n_c^*) + \phi_v(n_v^*) + \phi_c(n_c^*) \ \leq\ 2C(\mathbf{s}^*), \end{aligned}$$

	$c_1(\ell) \dots c_m(\ell)$	$c_{m+1}(\ell)$
$\ell = 0$	0	0
$\ell = 1$	1	1
$2 \leq \ell \leq m$	∞	1
$\ell \geq m + 1$	∞	∞

Table 1. The lower bound construction for budget-balanced resource-aware cost-sharing protocols

where for the first inequality, observe that if there were only n_v players, the optimal cost would be $\phi_v(n_v)$, since $n_v \leq n^{max}$. This means that, since by Claim 5.7 $n_v \geq n_v^*$, it holds that $\phi_v(n_v) \leq \phi_v(n_v^*) + \phi_c(n_v - n_v^*)$. □

6 MACHINES WITH GENERAL COST FUNCTIONS

In this section we show that the positive results of the previous sections do not carry over to general increasing cost functions. Remarkably, the instances that we construct in our PoA lower bounds use just capacitated constant cost functions, a seemingly mild extension beyond convex and concave functions.

We first focus on budget-balanced protocols and show that the PoA of any budget-balanced resource-aware cost-sharing protocol is $\Omega(n)$, even for unweighted players. It is not hard to check that the PoA is guaranteed to be $O(n)$ for any budget-balanced resource-aware cost-sharing protocol, with unweighted players,[3] so this bound is tight.

Theorem 6.1. *Every stable budget-balanced resource-aware cost-sharing mechanism has a PoA of $\Omega(n)$ for the class of capacitated constant cost functions.*

Proof. Consider a class of games comprising a set of $m + 1$ machines with costs as in Table 1. A resource-aware cost-sharing protocol for each one of these machines can depend arbitrarily on all this information, so even the first m machines can have a different protocol, although they have the same cost function. However, the cost-sharing cannot depend on the number, n, of jobs that are participating in the game.

First, assume that the number of jobs in the game is $n = 2m$. In this case, an optimal schedule assigns exactly one player to each of the first m machines and the remaining m players are assigned to the last machine. This allocation results in a total cost of $m + 1$, and any other allocation would lead to an infinite total cost. Since the cost-sharing protocol is stable, this game has at least one Nash equilibrium and, if any sub-optimal schedule is an equilibrium, then the PoA is unbounded. Otherwise, if an optimal schedule is an equilibrium, let S be the set of m players using the first m machines in this equilibrium. Then, none of them can decrease their cost by deviating to any other one of the first m machines.

Now, consider the different game from the same class where only the set S of players participate, i.e., $n = m$. Then, assigning them to the same machines that they were assigned to in the equilibrium of the previous game, would have to be an equilibrium for the new game: they do not want to deviate to each other's machines, and they also do not want to deviate to the unused machine since

[3] If the allocation in the pure Nash equilibrium is different from the optimal allocation, there should exist a machine j with n_j and n_j^* players in the equilibrium and in the optimal allocation, respectively, such that $n_j < n_j^*$. The cost of any player in machine j is upper bounded by $c_j(n_j) \leq c_j(n_j^*)$ and the cost of any other player is upper bounded by $c_j(n_j + 1) \leq c_j(n_j^*)$, because she has no benefit to deviate to j. $c_j(n_j^*)$ is trivially upper bounded by the cost of the optimal allocation and by summing up over all players the $O(n)$ bound follows.

their cost would remain the same, due to budget-balance. However, the optimal schedule in the new game would assign them all to the last machine, achieving a social cost of 1 instead of n. □

We now prove that the PoA of any resource-aware, not necessarily budget-balanced, cost-sharing protocol is $\Omega(\sqrt{n})$ even for unweighted players. We should emphasize that showing lower bounds for resource-aware mechanisms that allow overcharging is much more involved than the corresponding proofs for budget-balanced mechanisms, since the design space is considerably larger. In particular, we would like to direct the reader's attention to the second half of the following proof, which provides some intuition regarding the difficulty involved.

THEOREM 6.2. *Every stable resource-aware cost-sharing mechanism has a PoA of $\Omega(\sqrt{n})$ for the class of capacitated constant cost functions.*

PROOF. In the instances that we consider, $n = \Theta(m)$, so we just prove the bound as a function of m, and the same bound as a function of n is implied.

Consider the following family of problem instances involving $m + 1$ machines. The first group contains m machines that have a cost function $c_j(\ell)$, such that $c_j(1) = 1$ and $c_j(\ell) = \infty$ for $\ell > 1$. The second group contains a single machine with a cost of $c_{m+1}(\ell) = \sqrt{m}$ for $\ell \leq m$ and $c_{m+1}(\ell) = \infty$ for $\ell > m$. The resource-aware protocol can change the cost functions by increasing the induced costs that need to be shared among the agents, but the following case analysis shows that, no matter how the designer changes the shared costs, the price of anarchy of the induced game is $\Omega(\sqrt{m})$.

Let m' be the number of machines from the first group whose, possibly overcharged, cost functions are $\hat{c}_i(1) < \sqrt{m}$ and therefore, there exist $m + 1 - m'$ machines with cost at least $\sqrt{m}$. If $m' < m/2$, consider the instance in which the number of agents that are active is $n = 2m$. In this case, the only equilibrium is the one in which m players are matched to the first m machines, and the remaining m players are assigned to the last machine, leading to a social cost of at least $m' + (m + 1 - m')\sqrt{m} \in \Omega(m\sqrt{m})$. The same assignment prior to the overcharging costs though would yield a cost of $m + \sqrt{m}$, which implies that the price of anarchy would be $\Omega(\sqrt{m})$.

If, on the other hand $m' \geq m/2$, then consider the instance in which the number of agents that arrive at the system is $n = m'$. If all these agents were matched, in equilibrium, to one of the m' machines whose cost is less than $\sqrt{m}$, then this would also lead to a PoA of $\Omega(\sqrt{m})$. To verify this fact, note that the total cost of this assignment would be at least m', while the social optimum cost would be $\sqrt{m}$, if all of them chose the last machine. Therefore the price of anarchy for any such setting would be at least $m'/\sqrt{m} \in \Omega(\sqrt{m})$.

However, why this assignment of the m' agents to the m' machines of the first group would be an equilibrium is unclear. It is true that the cost of each player in this assignment would be at most $\sqrt{m}$, i.e., less than any other machine that is not occupied by another player. But, we cannot exclude the possibility of at least one of these players has an incentive to deviate to one of the other $m' - 1$ occupied machines. Such a deviation would lead to a cost of infinity on that machine, but the protocol could just charge all of this cost to the existing occupant, thus charging the new occupant 0! Such a cost-sharing protocol would then incentivize this player to deviate, in turn forcing the previous occupant to move to another machine. In fact, each machine could have a different ordering over which players to give higher priority to aiming to prevent any such equilibrium from arising.

So, how can we prove that for any stable resource-aware protocol there exists a set of m' players such that the assignment described above is an equilibrium? We consider what would happen if $n = 2m$ agents arrived instead. The crucial observation is that, since the protocol is stable, it *must* possess an equilibrium in this instance as well. The only equilibrium that may exist is the one in which m players are matched to the first m machines, and the remaining m players are assigned to

the last machine. Since this assignment is an equilibrium, the agents occupying the m' "cheaper" machines of the first group have no incentive to deviate to another one of these m' machines. Hence, if we remove everyone else, except these m' players, the remaining assignment needs to be an equilibrium. In particular, it needs to be an equilibrium because the protocol is resource-aware, so its cost sharing cannot depend on whether the other $2m - m'$ players exist or not. □

7 COST-SHARING FOR TWO MACHINES

We finally focus on the special case of instances comprising two machines and unweighted jobs. We show that for general cost-functions, there is a budget-balanced cost-sharing protocol with PoA = 2. Notably, this mechanism does not belong to the class of generalized weighted Shapley mechanisms [8], yet we show that it always induces Nash equilibria. Then we show a lower bound of 1.36 for resource-aware protocols with overcharging.

7.1 Budget-Balanced Cost-Sharing

Let $\alpha(\mathbf{s}) = \arg\min_i\{\pi(i)|i \in S_1(\mathbf{s})\}$ and $\beta(\mathbf{s}) = \arg\max_i\{\pi(i)|i \in S_2(\mathbf{s})\}$ be the first player of the first machine and the last player of the second machine in $\mathbf{s}$, with respect to the global ordering π.

Definition 7.1 (Increasing-Decreasing protocol). For any profile $\mathbf{s}$, player $\alpha(\mathbf{s})$ is charged the whole cost of the first machine and player $\beta(\mathbf{s})$ is charged the whole cost of the second. The rest of the players are charged 0:

$$\xi_{ij}(\mathbf{s}) = \begin{cases} c_j(\ell_j(\mathbf{s})), & \text{if } j = 1 \text{ and } i = \alpha(\mathbf{s}) \\ c_j(\ell_j(\mathbf{s})), & \text{if } j = 2 \text{ and } i = \beta(\mathbf{s}) \\ 0, & \text{otherwise.} \end{cases}$$

Theorem 7.2. *For instances with two machines, unweighted jobs, and general cost functions, the Increasing-Decreasing protocol is stable and it yields a PoA of* 2.

Proof. We verify that this protocol is stable by constructing a pure Nash equilibrium. Starting from a schedule where all the jobs are assigned to the first machine, while the first player (w.r.t. π) using machine 1 prefers to deviate to machine 2 we let him do so, until this is not the case. It is easy to verify that this schedule $\mathbf{s}$ is an equilibrium with n_1 jobs on the first machine and n_2 on the second, such that $c_1(n_1) \leq c_2(n_2 + 1)$ and $c_2(n_2) \leq c_1(n_1 + 1)$. The only two players suffering a cost are the last player to deviate (and he preferred to do so) and the first player who did not want to deviate (who also preferred not to). The only alternative choice of these two players is to deviate and suffer the full cost of the other machine.

Now, assume that the optimal allocation assigns n_1^* and n_2^* players to machines 1 and 2, respectively. The case of $n_1^* = n_1$ and $n_2^* = n_2$, trivially results in a PoS of 1. Without loss of generality, let $n_1^* > n_1$ and $n_2^* < n_2$. Then PoA ≤ 2 since,

$$c_1(n_1) + c_2(n_2) \;\leq\; c_1(n_1^*) + c_1(n_1 + 1) \;\leq\; 2c_1(n_1^*) \;\leq\; 2(c_1(n_1^*) + c_2(n_2^*)).$$

The example of Figure 1, with two players and two machines, shows that this bound is tight. Assume that $\pi(1) = 1$ and $\pi(2) = 2$. Then, the allocation where player 1 uses machine 2 and player 2 uses machine 1 is a Nash equilibrium. Indeed, if player 1 deviates to machine 1, he should pay 1, whereas he currently pays $1 - \varepsilon$. If player 2 deviates to machine 2, he should pay 2, whereas he currently pays 1. The optimal schedule would assign both players to machine 1, with a total cost 1. Hence, the PoA is at least $2 - \varepsilon$. For an arbitrarily small $\varepsilon > 0$. □

	$c_1(\ell)$	$c_2(\ell)$
$\ell = 0$	0	0
$\ell = 1$	1	$1 - \varepsilon$
$\ell = 2$	1	2

Fig. 1. Tight lower bound on the PoA of the Increasing-Decreasing protocol.

7.2 Cost-Sharing with Overcharging

We conclude by demonstrating that we cannot avoid losing a constant factor in games with just two machines, even if we used overcharging.

Theorem 7.3. *There is no stable overcharging resource-aware cost-sharing protocol that achieves PoA* $\leq (1+\sqrt{3})/2 \approx 1.36$*, even for instances with two machines with concave and convex cost functions.*

Proof. Consider two machines with the cost functions as shown in the Figure 2, where $c \in (0, 1/2)$ is some constant to be defined later.

	$c_1(\ell)$	$c_2(\ell)$
$\ell = 0$	0	0
$\ell > 0$	1	$2(\ell - 1) + c$

Fig. 2. Lower bound on the PoA of any resource-aware protocol, after overcharging. $0 < c < 1/2$.

Assume that two players appear and consider any budget-balanced protocol. Observe that the profile where both players choose the second machine is not a Nash equilibrium, since the cost would be more than 2 and, hence, at least one of the players would be paying more than 1. The profile where both players choose the first machine is not a Nash equilibrium either, since at least one of the players would be paying at least $\frac{1}{2}$, giving him reason to deviate to the second machine. Therefore, the only Nash equilibrium is the schedule where one player chooses the first machine and the other one chooses the second machine. The PoA then is $1 + c$.

The only way to achieve a better PoA for this case is to force the player that chooses the second machine to move to the first machine. In order to achieve this, the cost of the second machine for $\ell = 1$ should be raised at least to the value of $\frac{1}{2}$. But then, for instances where only one player appears the PoA becomes $\frac{1/2}{c}$. Choosing the value of c that satisfies the equality $1 + c = \frac{1}{2c}$, the PoA cannot be better than $1 + c = \frac{1+\sqrt{3}}{2}$ even with overcharging. □

REFERENCES

[1] Fidaa Abed and Chien-Chung Huang. 2012. Preemptive coordination mechanisms for unrelated machines. In *European Symposium on Algorithms*. Springer, 12–23.

[2] Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Eva Tardos, Tom Wexler, and Tim Roughgarden. 2008. The price of stability for network design with fair cost allocation. *SIAM J. Comput.* 38, 4 (2008), 1602–1623.

[3] Baruch Awerbuch, Yossi Azar, and Amir Epstein. 2005. The price of routing unsplittable flow. In *ACM Symposium on Theory of Computing*. ACM, 57–66.

[4] Yossi Azar, Lisa Fleischer, Kamal Jain, Vahab S. Mirrokni, and Zoya Svitkina. 2015. Optimal Coordination Mechanisms for Unrelated Machine Scheduling. *Operations Research* 63, 3 (2015), 489–500.

[5] Sayan Bhattacharya, Sungjin Im, Janardhan Kulkarni, and Kamesh Munagala. 2014. Coordination mechanisms from (almost) all scheduling policies. In *5th conference on Innovations in theoretical computer science*. ACM, 121–134.

[6] Kshipra Bhawalkar, Martin Gairing, and Tim Roughgarden. 2014. Weighted congestion games: Price of anarchy, universal worst-case examples, and tightness. *ACM Transactions on Economics and Computation* 2, 4 (2014), 14.
[7] Ioannis Caragiannis. 2013. Efficient coordination mechanisms for unrelated machine scheduling. *Algorithmica* 66, 3 (2013), 512–540.
[8] Ho-Lin Chen, Tim Roughgarden, and Gregory Valiant. 2010. Designing network protocols for good equilibria. *SIAM J. Comput.* 39, 5 (2010), 1799–1832.
[9] George Christodoulou, Elias Koutsoupias, and Akash Nanavati. 2009. Coordination mechanisms. *Theor. Comput. Sci.* 410, 36 (2009), 3327–3336.
[10] George Christodoulou, Stefano Leonardi, and Alkmini Sgouritsa. 2016. Designing Cost-Sharing Methods for Bayesian Games. In *Algorithmic Game Theory - 9th International Symposium, SAGT 2016*. 327–339.
[11] Giorgos Christodoulou, Kurt Mehlhorn, and Evangelia Pyrga. 2014. Improving the price of anarchy for selfish routing via coordination mechanisms. *Algorithmica* 69, 3 (2014), 619–640.
[12] George Christodoulou and Alkmini Sgouritsa. 2016. Designing Networks with Good Equilibria under Uncertainty. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016*. 72–89.
[13] Richard Cole, José R. Correa, Vasilis Gkatzelis, Vahab S. Mirrokni, and Neil Olver. 2015. Decentralized utilitarian mechanisms for scheduling games. *Games and Economic Behavior* 92 (2015), 306–326.
[14] Dimitris Fotakis and Paul G. Spirakis. 2008. Cost-balancing tolls for atomic network congestion games. *Internet Mathematics* 5, 4 (2008), 343–363.
[15] Martin Gairing and Florian Schoppmann. 2007. Total latency in singleton congestion games. In *Internet and Network Economics*. Springer, 381–387.
[16] Vasilis Gkatzelis, Konstantinos Kollias, and Tim Roughgarden. 2016. Optimal Cost-Sharing in General Resource Selection Games. *Operations Research* 64, 6 (2016), 1230–1238.
[17] Ragavendran Gopalakrishnan, Jason R. Marden, and Adam Wierman. 2014. Potential games are necessary to ensure pure Nash equilibria in cost sharing games. *Mathematics of Operations Research* (2014).
[18] Tobias Harks and Max Klimm. 2012. On the existence of pure Nash equilibria in weighted congestion games. *Mathematics of Operations Research* 37, 3 (2012), 419–436.
[19] Tobias Harks, Max Klimm, and Rolf H. Möhring. 2011. Characterizing the existence of potential functions in weighted congestion games. *Theory of Computing Systems* 49, 1 (2011), 46–70.
[20] Tobias Harks and Konstantin Miller. 2011. The worst-case efficiency of cost sharing methods in resource allocation games. *Operations Research* 59, 6 (2011), 1491–1503.
[21] Nicole Immorlica, Li Erran Li, Vahab S. Mirrokni, and Andreas S. Schulz. 2009. Coordination mechanisms for selfish scheduling. *Theoretical Computer Science* 410, 17 (2009), 1589–1598.
[22] Konstantinos Kollias. 2013. Nonpreemptive coordination mechanisms for identical machines. *Theory of Computing Systems* 53, 3 (2013), 424–440.
[23] Konstantinos Kollias and Tim Roughgarden. 2015. Restoring pure equilibria to weighted congestion games. *ACM Transactions on Economics and Computation* 3, 4 (2015), 13–46.
[24] Jason R. Marden and Adam Wierman. 2013. Distributed welfare games. *Operations Research* 61, 1 (2013), 155–168.
[25] Igal Milchtaich. 1996. Congestion games with player-specific payoff functions. *Games and Economic Behavior* 13, 1 (1996), 111–124.
[26] Dov Monderer and Lloyd S. Shapley. 1996. Potential games. *Games and Economic Behavior* 14, 1 (1996), 124–143.
[27] Damon Mosk-Aoyama and Tim Roughgarden. 2009. Worst-case efficiency analysis of queueing disciplines. In *International Colloquium on Automata, Languages and Programming*. Springer, 546–557.
[28] Hervé Moulin. 1999. Incremental cost sharing: Characterization by coalition strategy-proofness. *Social Choice and Welfare* 16, 2 (1999), 279–320.
[29] Hervé Moulin. 2008. The price of anarchy of serial, average and incremental cost sharing. *Economic Theory* 36, 3 (2008), 379–405.
[30] Hervé Moulin and Scott J. Shenker. 2001. Strategyproof sharing of submodular costs: budget balance versus efficiency. *Economic Theory* 18, 3 (2001), 511–533.
[31] Robert W. Rosenthal. 1973. The network equilibrium problem in integers. *Networks* 3, 1 (1973), 53–59.
[32] Philipp von Falkenhausen and Tobias Harks. 2013. Optimal cost sharing protocols for scheduling games. *Mathematics of Operations Research* 38, 1 (2013), 184–208.